I claim:

1. A method for enhancing pointer analysis, the method comprising:

processing an assignment between two variables in a program, wherein processing an assignment includes forming a relationship between two locations that are related to the two variables, wherein each location includes a label and a content, and wherein a content of one of the two locations is selectively unified with a content of an other of the two locations; and

propagating a label of the one of the two locations to a label of the other of the two locations such that the label of the one of the two locations is a subset of the other of the two locations.

2. The method of claim 1, wherein the act of propagating is delayed for a predetermined period of time so as to allow the act of processing an assignment to be executed for each assignment in the program.

3. The method of claim 1, further comprising forming a points-to graph by iterating the act of processing an assignment for each assignment in the program.

4. The method of claim 3, wherein forming a points-to graph includes forming a plurality of nodes, and forming a flow line between two nodes so as to represent the relationship between the two locations.

5. A computer readable medium having instructions stored thereon for causing a computer to perform a method for enhancing pointer analysis, the method comprising:

processing an assignment between two variables in a program, wherein processing an assignment includes forming a relationship between two locations that are related to the two variables, wherein each location includes a label and a content, and wherein a content of one of the two locations is selectively unified with a content of an other of the two locations; and

propagating a label of the one of the two locations to the label of the other of the two locations such that the label of the one of the two locations is a subset of the other of the two locations.

6. A method of analyzing pointers in a program, the method comprising:

forming a location for at least one variable in the program, wherein the location includes a label and a content; and

defining a relationship between two locations upon an assignment in the program, wherein a label of one of the two locations is associated with a label of an other of the two locations, and wherein contents of the two locations are selectively unified.

7. The method of claim 6, further comprising propagating the label of the one of the two locations to the other of the two locations so as to make the label of the one of the two locations a subset of the label of the other of the two locations.

8. The method of claim 6, wherein forming a location includes forming a location that points to another location, and wherein the another location defines a pointed-to location of the location.

9. The method of claim 8, further comprising defining at least one level, wherein the at least one level is defined by at least one location, wherein a pointed-to location of the at least one location defines another level.

10. The method of claim 9, wherein defining a relationship includes defining a relationship between the two locations that are in the same level.

11. The method of claim 10, wherein defining a relationship includes defining a relationship between the two locations that are in different levels.

12. A method of analyzing pointers in a program, the method comprising:

forming a location for at least one variable in the program, wherein the location includes a label and a content; and

forming a relationship between two locations upon an assignment of a first variable and a second variable in the program, wherein the relationship defines that a label of one of the two locations is a subset of a label of an other of the two locations, and wherein contents of the two locations are selectively unified.

13. The method of claim 12, wherein forming a location includes forming a location that points to another location, and wherein the another location defines a pointed-to location of the location.

14. The method of claim 13, wherein forming a relationship includes forming a relationship between two locations upon an assignment of a first variable and a second variable, wherein the second variable is assigned to the first variable.

15. The method of claim 14, wherein forming a location includes forming a first location for the first variable and forming a second location for the second variable, wherein the first location points to the other of the two locations, and wherein the second location points to the one of the two locations.

16. The method of claim 15, further comprising determining that the program is correctly typed given that the second variable is assigned to the first variable if and only if a label of a pointed-to location of the second location is a subset of a label of a pointed-to location of the first location, and wherein a content of the pointed-to

location of the first location is selectively unified with a content of the pointed-to location of the second location.

17. A method of analyzing pointers in a program, the method comprising:

forming a location for at least one variable in the program, wherein the location includes a label and a content; and

forming a relationship between two locations upon an assignment of a first variable and an address of a second variable in the program, wherein the relationship defines that a label of one of the two locations is a subset of a label of an other of the two locations, and wherein contents of the two locations are selectively unified.

18. The method of claim 17, wherein forming a location includes forming a location that points to another location, and wherein the another location defines a pointed-to location of the location.

19. The method of claim 18, wherein forming a relationship includes forming a relationship between two locations upon an assignment of a first variable and an address of the second variable, wherein the address of the second variable is assigned to the first variable.

20. The method of claim 19, wherein forming a location includes forming a first location for the first variable and forming a second location for the second variable, wherein the first location points to the other of the two locations, and wherein the second location is the one of the two locations.

21. The method of claim 20, further comprising determining that the program is correctly typed given that the address of the second variable is assigned to the first variable if and only if a label of the second location is a subset of a label of a pointed-to location of the first location, and wherein a content of the pointed-to

location of the first location is selectively unified with a content of the second location.

22. A method of analyzing pointers in a program, the method comprising:

forming a location for at least one variable in the program, wherein the location includes a label and a content; and

forming a relationship between two locations upon an assignment of a first variable and a dereference of a second variable in the program, wherein the relationship defines that a label of one of the two locations is a subset of a label of an other of the two locations, and wherein contents of the two locations are selectively unified.

23. The method of claim 22, wherein forming a location includes forming a location that points to another location, and wherein the another location defines a pointed-to location of the location.

24. The method of claim 23, wherein forming a relationship includes forming a relationship between two locations upon an assignment of a first variable and a dereference of a second variable, wherein the dereference of the second variable is assigned to the first variable.

25. The method of claim 24, wherein forming a location includes forming a first location for the first variable and forming a second location for the second variable, wherein the first location points to the other of the two locations, wherein the second location points to a first pointed-to location, and wherein the first pointed-to location points to the one of the two locations to define a second pointed-to location.

26. The method of claim 25, further comprising determining that the program is correctly typed given that the dereference of the second variable is assigned to the

first variable if and only if a label of the second pointed-to location is a subset of a label of a pointed-to location of the first location, and wherein a content of the pointed-to location of the first location is selectively unified with a content of the second pointed-to location.

27. A method of analyzing pointers in a program, the method comprising:

forming a location for at least one variable in the program, wherein the location includes a label and a content; and

forming a relationship between two locations upon an assignment of a dereference of a first variable and a second variable in the program, wherein the relationship defines that a label of one of the two locations is a subset of a label of an other of the two locations, and wherein contents of the two locations are selectively unified.

28. The method of claim 27, wherein forming a location includes forming a location that points to another location, and wherein the another location defines a pointed-to location of the location.

29. The method of claim 28, wherein forming a relationship includes forming a relationship between two locations upon an assignment of a dereference of a first variable and a second variable, wherein the second variable is assigned to the dereference of the first variable.

30. The method of claim 29, wherein forming a location includes forming a first location for the first variable and forming a second location for the second variable, wherein the first location points to a pointed-to location that points to the other of the two locations to defined a first pointed-to location, wherein the second location points to the one of the two locations.

31. The method of claim 30, further comprising determining that the program is correctly typed given that the second variable is assigned to the dereference of the first variable if and only if a label of a pointed-to location of the second location is a subset of a label of the first pointed-to location, and wherein a content of the first pointed-to location is selectively unified with a content of the pointed-to location of the second location.

32. A computer readable medium having instructions stored thereon for causing a computer to perform a method of analyzing pointers in a program, the method comprising:

forming a location for at least one variable in the program, wherein the location includes a label and a content; and

defining a relationship between two locations upon an assignment in the program, wherein a label of one of the two locations is defined as a subset of a label of an other of the two locations, and wherein contents of the two locations are selectively unified.

33. The method of claim 32, wherein defining a relationship includes defining a relationship between two locations upon an assignment of a first variable and a second variable in the program, wherein the first variable and the second variable are pointers.

34. The method of claim 32, wherein defining a relationship includes defining a relationship between two locations upon an assignment of a first variable and an address of a second variable in the program.

35. The method of claim 32, wherein defining a relationship includes defining a relationship between two locations upon an assignment of a first variable and a dereference of a second variable in the program.

36. The method of claim 32, wherein defining a relationship includes defining a relationship between two locations upon an assignment of a dereference of a first variable and a second variable.

37. A method of graphing variables in a program, the method comprising:

displaying a plurality of nodes, wherein each node of the plurality of nodes represent at least one variable;

displaying a plurality of lines, wherein at least one line of the plurality of lines represents a pointer relationship between two nodes; and

emanating a flow line from one node of the plurality of nodes to another node of the plurality of nodes so as to create a relationship between the one node and the another node when an assignment in the program is defined.

38. The method of claim 37, wherein displaying a plurality of lines includes display only one line that emanates from one node of the plurality of nodes.

39. The method of claim 37, wherein emanating a flow line includes emanating only one flow line from one node of the plurality of nodes.

40. A computer readable medium having instructions stored thereon for causing a computer to perform a method of graphing variables in a program, the method comprising:

displaying a plurality of nodes, wherein each node of the plurality of nodes represent at least one variable;

displaying a plurality of lines, wherein at least one line of the plurality of lines represents a pointer relationship between two nodes; and

emanating a flow line from one node of the plurality of nodes to another node of the plurality of nodes so as to associate the one node with the another node when an assignment in the program is defined.

41. A graph for enhancing pointer analysis, the graph comprising:

a plurality of nodes, wherein at least one node of the plurality of nodes represents at least one variable;

a plurality of lines, wherein at least one line of the plurality of lines represents a pointer relationship between at least two nodes of the plurality of nodes; and

at least one flow line, wherein the at least one flow line represents a label relationship between one node of the plurality of nodes and another node of the plurality of nodes.

42. The graph of claim 41, wherein the variable is adapted to be a pointer type through a process that is selected from a group consisting of declaring through a set of predefined data types, type conversion, and type casting.

43. The graph of claim 41, wherein only one line of the plurality of lines emanates from a node of the plurality of nodes.

44. The graph of claim 41, wherein only one flow line emanates from a node of the plurality of nodes.

45. The graph of claim 41, wherein the at least one flow line emanates from the one node of the plurality of nodes to the another node of the plurality of nodes so as to represent that a label of a location of the one node of the plurality of nodes is a subset of a label of a location of the another node of the plurality of nodes.

46. The graph of claim 41, wherein at least two nodes of the plurality of nodes are selectively unified.

47. A data structure to enhance pointer analysis in a program, wherein the program includes at least one assignment statement of variables, wherein each variable includes a name and a content, the data structure comprising:

a data member location; and

a data member flow to represent at least one label relationship.

48. The data structure of claim 47, wherein the data member location includes:

a data member label, wherein the data member label includes at least one data member symbol that represents a name of a variable; and

a data member content that represents a content of the variable or a unification of at least two variables.

49. The data structure of claim 48, wherein the data member flow stores an address of an instantiation of the data structure if an assignment statement is defined for two variables, and wherein the instantiation is related to one of the two variables.

50. The data structure of claim 49, wherein the data structure includes a method member propagate, wherein the method member propagate causes a propagation of the at least one data member symbol so as to make the data member label of one instantiation of the data structure a subset of a data member label of another instantiation of the data structure.

51. The data structure of claim 49, wherein the data structure includes a method member unify, wherein the method member unify merges a data member label of one instantiation of the data structure with a data member label of another instantiation of the data structure, and wherein the method member unify unifies a data member content of one instantiation of the data structure with a data member content of another instantiation of the data structure.

52. A method for enhancing pointer analysis, the method comprising:

processing a plurality of assignment statements in a program to derive a plurality of sets of information, wherein the plurality of sets of information is distributed among a plurality of levels of indirection; and

unifying selectively sets of information in at least one level of indirection so as to allow a desired level of analytical precision within a desired duration of pointer analysis.

53. The method of claim 52, wherein the act of unifying includes unifying sets of information in all levels of indirection except for a first level of indirection.

54. The method of claim 52, wherein the desired duration of pointer analysis is about linearly proportional to the size of the program.

55. A system for enhancing pointer analysis of a program, wherein the program includes at least one source file, the system comprising:

a compiler to compile the at least one source file to produce an intermediate language;

a builder receptive to the intermediate language to build a tree that represents the at least one source file; and

an analyzer to analyze the tree to produce an object file, wherein the object file contains at least one relationship between two variables in an assignment statement in the program, wherein the relationship defines that a set of symbols relating to one of the two variables is a subset of a set of symbols relating to an other of the two variables.

56. The system of claim 55, further comprising a linker to link a plurality of object files of the program so as to produce a pointer analysis for the program.